Sous-structuration et méthode asymptotique numérique en élasticité non linéaire

Mustapha Essakhi[*] — Bouazza Braikat[*]
Hassane Lahmam[*] — Jean-Marc Cadou[**]
Noureddine Damil[*] — Michel Potier-Ferry[**]

[*] *Laboratoire de Calcul Scientifique en Mécanique,*
Faculté des Sciences Ben M'Sik, Université Hassan II - Mohammedia,
Sidi Othman, Casablanca, Maroc
saki_m@caramail.com, n.damil@univh2m.ac.ma

[**] *Laboratoire de Physique et Mécanique des Matériaux, UMR CNRS 7554,*
I.S.G.M.P., Université de Metz, Ile du Saulcy, 57045 Metz cedex 01, France
potier-ferry@lpmm.univ-metz.fr

RÉSUMÉ. *On se propose dans cet article de coupler la Méthode Asymptotique Numérique (MAN) avec une technique de sous-structuration et un calcul explicite de la matrice d'interface. De cette manière, on obtient une décomposition exacte de la matrice tangente, qui est souhaitable puisque la MAN implique une résolution de nombreux problèmes linéaires avec une même matrice. Le coût de calcul de cette condensation est analysé et comparé au coût de traitement des seconds membres de la MAN. Des exemples sont analysés, issus de modèles de post-flambage de plaques longues.*

ABSTRACT. *In this paper, a coupling between Asymptotic Numerical Methods (ANM) and a technique of substructures is presented, with an explicit calculation of the interface stiffness matrix. In this way, an exact triangulation of the tangent stiffness is obtained. This is suitable because ANM implies to solve many linear problems with a single matrix. The computational cost of this condensation is analysed and compared with the cost of the treatement of the right-hand sides of ANM. Some examples are analysed, emanating from post-buckling modelling of long plates.*

MOTS-CLÉS : *méthode asymptotique numérique, sous-structuration, non linéaire géométrique, bande mince.*

KEYWORDS: *asymptotique numerical method, substructure, geometric non linearity, thin band.*

Revue européenne des éléments finis. Volume 12 - n° 4/2003, pages 407 à 426

1. Introduction

La résolution des problèmes non linéaires de grande dimension est difficilement réalisable avec des ordinateurs à architecture séquentielle. En effet de telles résolutions conduisent à des temps de calculs prohibitifs et demandent d'énormes capacités de stockage des données.

Cependant le passage à des problèmes à grand nombre de degrés de liberté nécessite le développement de nouvelles méthodes permettant de réduire le temps de calcul. En effet, les algorithmes de type Newton-Raphson ou Méthode Asymptotique Numérique (MAN) [COC 941], adaptés essentiellement aux machines séquentielles, ne répondent plus aux besoins impératifs que nécessite la résolution de systèmes de grande taille.

La méthode de Newton-Raphson est une méthode de type prédiction correction. La solution est obtenue point par point. Cette méthode est robuste et fiable, mais elle est coûteuse en temps CPU. Quant à la Méthode Asymptotique Numérique, elle se base sur les techniques de perturbation et de discrétisation.
L'idée de base de cette méthode est de chercher la solution en générant non pas une séquence de points mais une séquence de branches à la différence de la méthode de prédiction-correction. Chaque branche est représentée à l'aide d'une série, dont les termes vérifient des problèmes linéaires qui seront ensuite résolus par une procédure directe. Quand ces problèmes ont une grande taille, leurs résolutions deviennent plus difficiles. Parmi les méthodes de résolution adaptées à ce genre de problèmes, on trouve les méthodes de sous-structuration [PRE 63] et les méthodes de multigrilles [WES 92].

Les méthodes de sous-structurations ont été développées avec l'apparition des ordinateurs à architecture parallèle dès le début des années soixante et ont fait l'objet de nombreux travaux depuis cette époque [BAR 99], [ESC 92], [DES 90]. Elles consistent, comme leurs noms l'indiquent, à décomposer le problème initial en une succession de sous-problèmes locaux posés sur chaque sous-structure. Il existe plusieurs familles de méthodes de décomposition de domaines pour résoudre des problèmes non linéaires discrétisés par éléments finis. Parmis ces méthodes, on peut distinguer deux grandes classes : les méthodes avec recouvrement des sous-structures dont la plus connue est la méthode de Schwarz [SCH 69] et les méthodes de sous-structuration sans recouvrement [ESC 99]. La difficulté technique de ces méthodes réside dans le raccord au niveau des interfaces des sous-structures. Parmi les méthodes sans recouvrement les plus utilisées, on trouve les méthodes du complément de Schur primal et dual. La première consiste à imposer la continuité des déplacements dans la résolution des problèmes locaux et à se ramener à un problème au niveau de l'interface pour vérifier la continuité des contraintes. La deuxième consiste cette fois-ci à imposer la continuité des contraintes normales dans la résolution des problèmes locaux et à se ramener de la même manière à un problème d'interfaces permettant d'assurer la continuité des déplacements. Deux choix sont possibles pour la résolution du problème

primal ou dual d'interfaces [TAL 93] ; la plupart des auteurs préfèrent les résolutions itératives afin d'éviter le calcul explicite de la matrice d'interface [DAD 80].

Dans ce travail, on propose un algorithme basé sur l'association de la MAN et la sous-structuration. Son principe suit les étapes suivantes : décomposition du domaine, écriture de la formulation variationnelle des problèmes sur chaque sous-domaine, application de la technique de perturbation, discrétisation des problèmes linéaires obtenus, construction de la matrice d'interfaces, résolution des problèmes d'interfaces et calcul de restitution.

La résolution des problèmes d'interfaces peut se faire par un solveur itératif ou direct. La plupart des articles récents utilisent des solveurs itératifs. Mais dans la MAN, nous avons 15 ou 20 problèmes linéaires à résoudre avec la même matrice, ce qui *a priori* n'incite pas au choix d'un solveur itératif. Néanmoins des solveurs itératifs pour seconds membres répétés ont été proposés dans les références [FAR 942] et [REY 96] pour résoudre les problèmes linéaires issus d'un algorithme de type Newton ou dans la reférence [GAL 00] pour résoudre les problèmes linéaires issus d'une MAN. Ces algorithmes demandent le stockage à chaque résolution des directions de descente et la projection de ces directions de descente. Des étapes de réorthogonalisation de ces directions de descente sont nécessaires pour assurer la convergence de l'algorithme.

Une autre voie, plus simple et plus naturelle, consiste à calculer explicitement la matrice d'interface et c'est ce que nous explorerons dans cet article. Deux difficultés seront analysées en détail. La première consiste à limiter le coût du calcul des seconds membres, la seconde à construire la matrice d'interface de la manière la moins chère possible. Des exemples issus du flambage des plaques longues, permettront d'évaluer l'efficacité de la démarche proposée.

2. Problème à résoudre

Pour illustrer l'algorithme proposé, on considère un solide élastique homogène occupant un domaine Ω sous forme de bande, de frontière $\partial\Omega$ et soumis à des sollicitations extérieures surfaciques λF et éventuellement volumiques λg où λ est un paramètre de chargement.

La formulation variationnelle traduisant l'équilibre et la loi de comportement peuvent s'écrire sous la forme :

$$\begin{array}{rl} \int_\Omega S : \delta\gamma(u) d\Omega & = \lambda \int_\Omega g.\delta u d\Omega + \lambda \int_{\partial\Omega} F\ \delta u ds \\ S & = D : \gamma(u), \end{array} \qquad [1]$$

où D est le tenseur d'élasticité du matériau, S est le deuxième tenseur des contraintes de Piola-Kirchhoff. Le tenseur des déformations de Green-Lagrange γ et sa variation $\delta\gamma$ sont reliés au vecteur déplacement u par les relations suivantes :

$$\gamma = \gamma^l(u) + \gamma^{nl}(u,u) \quad et \quad \delta\gamma = \gamma^l(\delta u) + 2\gamma^{nl}(u,\delta u). \qquad [2]$$

Il s'agit de déterminer les déplacements u et les contraintes S vérifiant le problème non linéaire [1].

La démarche proposée, dans ce travail, pour le couplage de la Méthode Asymptotique Numérique et la méthode de sous-structuration est la suivante :

1- On décompose le domaine Ω en N sous-structures $\Omega(n)\ \ n = 1, N$.

2- On écrit l'équilibre et le comportement sur chaque sous-structure, on obtient ainsi N problèmes non linéaires.

3- On applique une technique de perturbation à l'ordre p, ce qui permet de transformer chacun des N problèmes non linéaires en p problèmes linéaires sur chaque sous-structure $\Omega(n)$.

4- On discrétise, par éléments finis, les $N.p$ problèmes linéaires

5- On construit la matrice condensée sur les $N + 1$ interfaces.

6- On résoud les p problèmes condensés. On décompose une seule fois la matrice condensée qui est la même pour tous les ordres. A chaque ordre i, on calcule un second membre qui dépend de la solution aux ordres précédents et on fait des montées-descentes puis une restitution. Et enfin, on construit la solution globale à l'ordre i.

3. Décomposition du domaine et écriture des problèmes à résoudre

On commence d'abord par décomposer la structure Ω en N sous-structures disjointes $\Omega(n)\ \ (n = 1, ...N)$. Chaque sous-structure est reliée avec les deux autres sous-structures voisines $\Omega(n-1)$ et $\Omega(n+1)$ par deux interfaces $I(n-1)$ et $I(n)$ comme indiqué sur la figure 1.

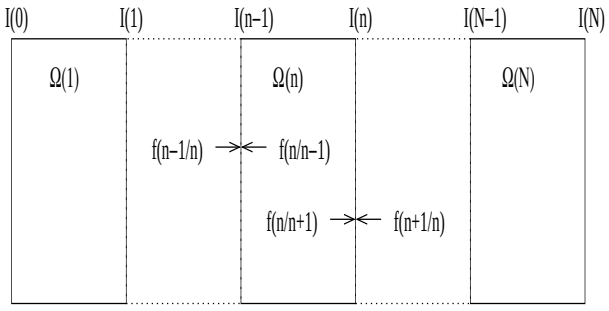

Figure 1. *La sous-structure $\Omega(n)$ et les interfaces $I(n-1)$ et $I(n)$ correspondantes*

On écrit ensuite la formulation variationnelle traduisant l'équilibre et le comportement sur chaque sous-structure $\Omega(n)$ pour $n = 1, N$, on obtient :

$$\begin{array}{rl} \int_{\Omega(n)} S : \delta\gamma(u) d\Omega & = \lambda \int_{\Omega(n)} g.\delta u d\Omega + \int_{I(n)} f((n+1)/n).\delta u ds \\ & + \int_{I(n-1)} f((n-1)/n).\delta u ds + \lambda \int_{\partial\Omega(n)} F\ .\delta u ds \\ S & = D : \gamma(u), \end{array} \quad [3]$$

où $f((n-1)/n)$ et $f((n+1)/n)$ sont les actions inconnues des sous-structures $\Omega(n-1)$ et $\Omega(n+1)$ sur la sous-structure $\Omega(n)$. On introduit les notations suivantes :

$$\begin{array}{ll} \langle f^{n-1,n}, \delta u\rangle & = \int_{I(n-1)} f((n-1)/n).\delta u ds \\ \langle f^{n+1,n}, \delta u\rangle & = \int_{I(n)} f((n+1)/n).\delta u ds \\ \langle F^n, \delta u\rangle & = \int_{\Omega(n)} g.\delta u d\Omega + \int_{\partial\Omega(n)} F\ .\delta u ds. \end{array} \quad [4]$$

Il s'agit de déterminer les déplacements u et les contraintes S des N problèmes non linéaires [3], ainsi que le paramètre de chargement λ. Pour cela, nous allons résoudre ces N problèmes non linéaires [3] par un algorithme de type asymptotique numérique.

3.1. *Application de la MAN aux problèmes locaux*

Nous allons appliquer la technique de perturbation simultanément aux N problèmes non linéaires [3]. On cherche alors les inconnues sous la forme d'une représentation en séries entières par rapport à un paramètre a, qui sera défini par la suite, autour d'une solution connue ($u_0, S_0, f_0^{n-1,n}$, $f_0^{n+1,n}, \lambda_0$).

Les inconnues $u, S, f^{n-1,n}, f^{n+1,n}$ et le paramètre de chargement λ, sur chaque sous-structure $\Omega(n)$, sont alors représentées par :

$$\left\{ \begin{array}{llllll} u & = u_0 & +a u_1 & +..... & +a^p\ u_p \\ S & = S_0 & +a S_1 & +..... & +a^p\ S_p \\ f^{n-1,n} & = f_0^{n-1,n} & +a f_1^{n-1,n} & +..... & +a^p f_p^{n-1,n} \\ f^{n+1,n} & = f_0^{n+1,n} & +a f_1^{n+1,n} & +..... & +a^p f_p^{n+1,n} \\ \lambda & = \lambda_0 & +a\ \lambda_1 & +..... & +a^p\ \lambda_p. \end{array} \right. \quad [5]$$

En injectant ces développements dans les équations non linéaires [3], nous obtenons une suite récurrente de problèmes linéaires vérifiés par les termes $U_k = (u_k, S_k)$, $f_k^{n-1,n}$, $f_k^{n+1,n}$, λ_k, pour chaque sous-structure et pour chaque ordre :

$$L_{tn}(U_k) = \lambda_k\ F^n + f_k^{n-1,n} + f_k^{n+1,n} + FQ_k^n \quad n = 1, N \ \ et \ \ k = 1, p, \quad [6]$$

où L_{tn} est l'opérateur tangent de la sous-structure $\Omega(n)$ qui ne dépend que du point de départ $U_0 = (u_0, S_0)$, FQ_k^n est le vecteur second membre à l'ordre k qui ne

dépend que des solutions aux ordres précédents pour chaque sous-structure $\Omega(n)$, F^n représente les actions données sur la sous-structure $\Omega(n)$ et le couple ($f_k^{n-1,n}$, $f_k^{n+1,n}$) représente les actions inconnues des sous-structures $\Omega(n-1)$ et $\Omega(n+1)$ sur $\Omega(n)$. Nous verrons, dans l'étape de condensation, comment éliminer ces actions inconnues en utilisant le principe de l'action et de la réaction.

3.2. *Techniques de discrétisation et de condensation*

L'élimination des contraintes et la discrétisation des $N.p$ problèmes linéaires [6], obtenus par la technique de perturbation, conduisent aux problèmes suivants :

$$[K_t^n]\{v_k\} = \lambda_k\{F^n\} + \{f_k^{n-1,n}\} + \{f_k^{n+1,n}\} + \{FQ_k^n\}\ n = 1, N\ \ k = 1, p. \quad [7]$$

où $\{v_k\}$ est la discrétisation du déplacement u_k. Dans le paragraphe suivant, nous allons montrer comment éliminer les actions inconnues $f_k^{n-1,n}$ et $f_k^{n+1,n}$ en utilisant le principe de l'action et de la réaction. Notons que la définition des problèmes [6] et [7] est identique à celle utilisée dans [COC 942], aux efforts d'interface près.

Avant de passer à l'étape de condensation, réécrivons la matrice $[\,K_t^n]$ en numérotant pour chaque sous-structure en premier les degrés de liberté de la première interface, indicés $I(n-1)$, puis les degrés de liberté internes, indicés $\Omega(n)$, et enfin les degrés de liberté de la deuxième interface, indicés $I(n)$ (voir figure 2).

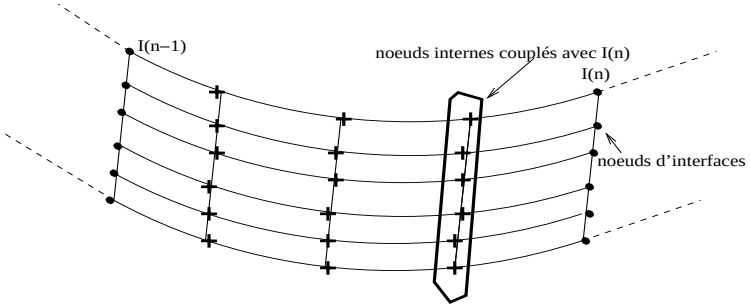

Figure 2. *La sous-structure $\Omega(n)$: nœuds internes $(+)$ et nœuds d'interfaces $(.)$*

De cette manière les problèmes linéaires [7] s'écrivent sous la forme suivante :

$$[K_t^n]\{v_k\} = \begin{bmatrix} [K_I^{n-1,n}] & [\,K_c^{n-1,n}] & [0] \\ [\,K_c^{n-1,n}]t & [\,K_\Omega^n] & [\,K_c^{n,n}] \\ [0] & [\,K_c^{n,n}]t & [\,K_I^{n,n}] \end{bmatrix} \begin{Bmatrix} v_{Ik}^{n-1} \\ v_{\Omega k}^n \\ v_{Ik}^n \end{Bmatrix}$$

$$= \lambda_k \begin{Bmatrix} F_I^{n,n-1} \\ F_\Omega^n \\ F_I^{n,n} \end{Bmatrix} + \begin{Bmatrix} f_k^{n-1,n} \\ 0 \\ f_k^{n+1,n} \end{Bmatrix} + \begin{Bmatrix} FQ_{Ik}^{n,n-1} \\ FQ_{\Omega k}^n \\ FQ_{Ik}^{n,n} \end{Bmatrix} \quad [8]$$

$$n = 1, .., N \quad et \quad k = 1, .., p.$$

La forme particulière des problèmes [8] est due au fait que nous avons choisi de décomposer la structure en sous-structures ayant chacune deux interfaces différentes. Cette forme fait intervenir deux blocs de couplage : $[\ K_c^{n-1,n}]$ correspond à la matrice de couplage qui combine les nœuds de l'interface $I(n-1)$ et les nœuds internes de $\Omega(n)$ et le bloc $[\ K_c^{n,n}]$ correspond à la matrice de couplage qui combine les nœuds de l'interface $I(n)$ et les nœuds internes de $\Omega(n)$. La diagonale contient les deux matrices $[\ K_I^{n-1,n}]$ et $[\ K_I^{n,n}]$ où n'interviennent respectivement que les nœuds des interfaces $I(n-1)$ et $I(n)$, ainsi que finalement la matrice $[\ K_\Omega^n]$ qui elle ne fait intervenir que les nœuds internes à $\Omega(n)$.

Dans le premier terme du second membre de l'expression [8], $\{F_I^{n,n-1}\}$, $\{F_\Omega^n\}$ et $\{F_I^{n,n}\}$ représentent les actions données appliquées respectivement sur l'interface $I(n-1)$, sur l'intérieur $\Omega(n)$ et sur l'interface $I(n)$. Dans le troisième terme de l'expression [8], $\{FQ_{Ik}^{n,n-1}\}$, $\{FQ_{\Omega k}^n\}$ et $\{FQ_{Ik}^{n,n}\}$ représentent les vecteurs seconds membres de la MAN correspondant respectivement à l'interface $I(n-1)$, à l'intérieur $\Omega(n)$ et à l'interface $I(n)$.

Remarquons que les blocs constituant la matrice tangente $[K_t^n]$, correspondant à la sous-structure $\Omega(n)$, ne dépendent pas de l'ordre de troncature de la série, contrairement aux seconds membres qui eux dépendent des solutions aux ordres précédents. C'est l'intérêt d'utiliser la Méthode Asymptotique Numérique.

La deuxième équation de [8] permet d'exprimer les inconnues associées à l'intérieur de la sous-structure $\Omega(n)$, à chaque ordre, $\{v_{\Omega k}^n\}$ en fonction des inconnues associées aux deux interfaces $\{v_{Ik}^{n-1}\}$ et $\{v_{Ik}^n\}$. En reportant cette relation dans la première et la troisième équation dans [8], on obtient deux équations exprimant les variables d'interface $\{v_{Ik}^n\}$ et $\{v_{Ik}^{n-1}\}$ en fonction des différents blocs de matrices, des différents vecteurs et des actions inconnues $\{f_k^{n-1,n}\}$ et $\{f_k^{n+1,n}\}$.

Pour éliminer ces actions inconnues, on écrit d'abord l'équilibre de deux sous-structures adjacentes $\Omega(n-1)$ et $\Omega(n+1)$, et ensuite le principe de l'action et de la réaction au niveau de chaque interface. En injectant le développement en série [5] dans le principe de l'action et de la réaction, on en déduit des relations à chaque ordre k. Par exemple pour l'interface $I(j)$ on obtient à l'ordre k :

$$\{f_k^{j+1,j}\} + \{f_k^{j,j+1}\} = 0. \qquad [9]$$

Pour éliminer l'action inconnue $\{f_k^{n-1,n}\}$ on utilise la première équation traduisant l'équilibre de $\Omega(n)$, la deuxième équation traduisant l'équilibre de $\Omega(n-1)$ et la relation [9] correspondant à l'interface $I(n-1)$. De même $\{f_k^{n+1,n}\}$ s'élimine en utilisant la deuxième équation traduisant l'équilibre de $\Omega(n)$, la première équation dans l'équilibre de $\Omega(n+1)$ et la relation [9] à l'interface $I(n)$. On obtient ainsi :

$$[K_c^{*(n-1,n)}]^t\{v_{Ik}^{n-1}\} + [K_I^{*(n)}]\{v_{Ik}^n\}\} + [K_c^{*(n,n+1)}]\{v_{Ik}^{n+1}\} = g_{Ik}^{*(n)}, \qquad [10]$$

avec

$$\left\{\begin{array}{ll} [K_c^{*(n-1,n)}]^t & = -[K_c^{n,n}]^t[K_\Omega^n]^{-1}[K_c^{n-1,n}]^t \\ \\ [K_I^{*(n)}] & = [\,K_I^{n,n+1}] - [\,K_c^{n,n+1}][\,K_\Omega^{n+1}]^{-1}[\,K_c^{n,n+1}]^t \\ & +[K_I^{n,n}] - [K_c^{n,n}]^t[\,K_\Omega^n]^{-1}[K_c^{n,n}] \\ \\ [\,K_c^{*(n,n+1)}] & = -[\,K_c^{n,n+1}][\,K_\Omega^{n+1}]^{-1}[\,K_c^{n+1,n+1}] \\ \\ g_{Ik}^{*(n)} & = [\,K_c^{n,n}]^t[\,K_\Omega^n]^{-1}\big(\lambda_k\{F_\Omega^n\} + \{FQ_{\Omega\,k}^n\}\big) \\ & -[\,K_c^{n,n+1}][\,K_\Omega^{n+1}]^{-1}\big(\lambda_k\{F_\Omega^{n+1}\} + \{FQ_{\Omega\,k}^{n+1}\}\big) \\ & +\lambda_k\big(\{F_I^{n,n}\} + \{F_I^{n+1,n}\}\big) + \{FQ_{Ik}^{n+1,n}\} + \{FQ_{Ik}^{n,n}\}. \end{array}\right. \qquad [11]$$

On remarque, dans la relation [11], que la matrice $[\,K_c^{*(n,n+1)}]$ est égale à la matrice $[K_c^{*(n-1,n)}]^t$ pour $n = n+1$ ($[K_\Omega^n]$ est symétrique). Ce qui nous permettra de minimiser les coûts de calcul pour la construction de la matrice d'interfaces.

En écrivant la relation [10] pour toutes les interfaces de $I(0)$ jusqu'à $I(N)$, on obtient finalement la matrice condensée assemblée sur toutes les interfaces :

$$[K_I^*] \quad \{v_{Ik}\} \quad = \{G_{Ik}^*\}. \qquad [12]$$

On a ainsi écrit les problèmes linéaires assemblés pour les degrés de liberté sur toutes les interfaces. Ce problème permet de déterminer tous les termes de la série pour les degrés de liberté sur toutes les interfaces. Dans l'équation [12], la matrice $[K_I^*]$ est la même pour tous les ordres, seuls les seconds membres assemblés $\{G_{Ik}^*\}$ changent. Remarquons que $[K_I^*]$ est tridiagonale par bloc (voir annexe, relation [20]). Une fois le problème [12] résolu, la relation ([17], annexe) permet de déterminer les déplacements associés aux degrés de liberté internes à la sous-structure $\Omega(n)$.

On peut remarquer que les problèmes [12] demandent une relation supplémentaire, celle du pilotage. Dans cette étude nous nous sommes limités à des pilotages à charge imposée, ce qui se traduit par la relation $a = \lambda - \lambda_0 = \epsilon\lambda_1 + ...\epsilon^p\lambda_p$. On en déduit alors les relations supplémentaires donnant les paramètres λ_k dans le second membre de la relation [12], soient :

$$\lambda_1 = 1 \qquad et \qquad \lambda_k = 0 \qquad pour \qquad k = 2, ...p. \qquad [13]$$

3.3. *Technique de continuation*

Lors du calcul de la branche solution d'un problème non linéaire [COC 941] par une MAN, une procédure efficace de continuation a été proposée. Il s'agit d'appliquer la méthode de perturbation proposée dans ce travail pas à pas. Pour cela, il suffit de définir un nouveau point de départ (u_0, S_0) à l'intérieur du domaine de convergence de la série et de réappliquer les développements [5] à partir de ce nouveau point pour

progresser le long de la branche cherchée. La représentation analytique de la branche non linéaire a permis d'analyser le domaine de convergence et de définir un critère simple [COC 942]. L'approximation polynomiale [5] est alors considérée valable sur l'intervalle $[0, a_{max}]$ où a_{max} est calculé par la formule suivante :

$$a_{max} = (\epsilon \frac{\| u_1 \|}{\| u_p \|})^{1/p-1}, \qquad [14]$$

où ϵ est une tolérance donnée et $\| \cdot \|$ est la norme euclidienne. Une fois a_{max} calculé, le nouveau point de départ est alors $u_0 = u(a_{max})$, $S_0 = S(a_{max})$ et $\lambda_0 = \lambda(a_{max})$.

4. Précisions sur le calcul de la matrice condensée et sur la résolution des problèmes [12]

Dans cette section, nous allons donner les étapes utilisées dans notre approche afin d'optimiser le calcul de $[K_I^*]$.

Premièrement, les différentes matrices $[K_c^{n-1,n}]$, $[K_c^{n,n}]$, $[K_I^{n-1,n}]$, $[K_I^{n,n}]$ et $[K_\Omega^n]$ sont créées directement par assemblage des matrices élémentaires des éléments qui constituent chaque sous-structure $\Omega(n)$. Ceci demande une renumérotation des degrés de liberté, afin de préciser les degrés de libertés internes, ceux d'interfaces et les degrés de liberté de couplages.

Ensuite, il faut calculer les matrices condensées $[K_c^{*(n-1,n)}]^t$ et $[K_I^{*(n)}]$. Il est inutile de calculer les matrices de couplages $[K_c^{*(n,n+1)}]$ qui sont égales aux matrices $[K_c^{*(n+1,n+2)}]^t$, voir formule [11].

Afin de calculer les 2 matrices $[P_1]$ et $[P_2]$, définies par :

$$\begin{cases} [P_1] & = [K_\Omega^n]^{-1}[K_c^{n,n}] \\ [P_2] & = [K_\Omega^{n+1}]^{-1}[K_c^{n,n+1}]^t, \end{cases} \qquad [15]$$

on factorise la matrice $[K_\Omega^n]$, puis on résout N_{dim} systèmes linéaires de la forme $[K_\Omega^n]\{q\} = \{X\}$ où $\{X\}$ est la i$^{\text{ième}}$ colonne de $[K_c^{n,n}]$. N_{dim} est le nombre de colonnes de $[K_c^{n,n}]$ et représente à peu près la largeur de bande dans $[K_\Omega^n]$ et dans $[K_I^*]$. Une fois $[P_1]$ et $[P_2]$ calculées, les blocs de la matrice d'interface condensée sont donnés par :

$$\begin{cases} [K_c^{*(n-1,n)}]^t & = -[P_1]^t[K_c^{n-1,n}]^t \\ [K_I^{*(n)}] & = [\ K_I^{n,n+1}] - [\ K_c^{n,n+1}][P_2] + [K_I^{n,n}] - [K_c^{n,n}]^t[P_1]. \end{cases} \qquad [16]$$

Les trois produits matrice-matrice dans [16] sont le produit de $[P_1]$ ou $[P_2]$ par l'une des matrices de couplage. Ces matrices de couplage sont creuses et afin d'optimiser ces produits nous avons utilisé un stockage de type morse ou compact. Cela revient à ne stocker que les coefficients non nuls des matrices de couplage et évite ainsi des multiplications inutiles qui conduisent à des temps de calcul importants.

Remarquons que les matrices $[P_1]$ et $[P_2]$ sont des matrices pleines rectangulaires ayant pour dimensions le nombre de degrés de liberté internes par le nombre de degrés de liberté d'interfaces. Ces matrices seront utilisées pour calculer les seconds membres $\{g_{Ik}^{*(n)}\}$ (voir formule [11]), ainsi que pour la restitution. Elle sont sauvegardées sur disque afin de diminuer l'occupation en mémoire.

Le point le plus délicat pour un calcul explicite de la matrice condensée est le calcul des matrices $[P_1]$ et $[P_2]$ qui nécessite N_{dim} montées-descentes à partir de la triangulation des matrices de sous-domaine. La procédure utilisée ici est analogue à celle proposée par [ESC 92]. Le coût de ce calcul pourrait être diminué fortement en cas de sous-structures identiques [MEO 02], mais pour un problème non linéaire, cet avantage ne pourrait servir que pour le premier pas de calcul.

5. Résumé de l'algorithme proposé

L'algorithme proposé consiste en trois phases principales : la phase de construction de la matrice d'interface dite aussi condensation des nœuds internes, la phase de décomposition de la matrice d'interface et enfin la phase de calcul des seconds membres qui elle, comprend une résolution du problème d'interface et une restitution de la solution aux nœuds internes, à chaque ordre de la série (voir figure 3).

6. Nombre d'opérations de l'algorithme proposé

Dans l'algorithme proposé, l'étape la plus importante est la construction de la matrice d'interfaces $[K_I^*]$. Soient N_{dim} la largeur de bande de la matrice $[K_\Omega^n]$, N_Ω le nombre de degrés de liberté des nœuds internes, N_I le nombre de degrés de liberté des nœuds d'une interface, β le facteur de remplissage des matrices de couplages et N le nombre de sous-structures.

La construction de $[K_I^*]$ demande N factorisations de $[K_\Omega^n]$, $(N+1)$ produits $[P_1]$ et $[P_2]$ ce qui revient à faire $2(N+1) * N_I$ montées-descentes et enfin $3(N+1)$ produits d'une matrice de couplage par une matrice $[P_1]$ ou $[P_2]$. Les nombres des opérations effectuées sont :
la factorisation de la matrice $[K_\Omega^n]$ soit :

$$N * N_\Omega * N_{dim}^2,$$

le calcul des produits $[P_1]$ et $[P_2]$ soit :

$$4(N+1) * N_I * N_\Omega * N_{dim},$$

le calcul du produit de $[P_1]$ ou $[P_2]$ par une matrice de couplage. Ces dernières sont creuses, chaque ligne ne contient que $\beta * N_I$ termes non nuls où β est donné par la relation suivante :

$$\beta = \frac{(NNPE-1) * (NDPN)}{N_I}$$

étape 1 *Calcul de la matrice d'interface* $[K_I^*]$

BOUCLE SUR LES SOUS-STRUCTURES n=1,N
Calcul de $[K_c^{n-1,n}], [K_c^{n,n}], [K_I^{n-1,n}], [K_I^{n,n}], [K_\Omega^n]$
Calcul de $[P^1], [P^2], [K_c^{*(n-1,n)}]^t, [K_I^{*(n)}], [K_c^{*(n,n+1)}]$
Construction de la matrice d'interface $[K_I^*]$
Fin de la BOUCLE SUR LES SOUS-STRUCTURES n=1,N

étape 2 *Décomposition de la matrice d'interface* $[K_I^*]$

étape 3 *Calcul des seconds membres + résolutions*

BOUCLE SUR LES ORDRES k=1,p
BOUCLE SUR LES SOUS-STRUCTURES n=1,N
Calcul de $g_{Ik}^{*(n)}$
Construction de $[G_{Ik}^*]$
Fin de la BOUCLE SUR LES SOUS-STRUCTURES n=1,N
montées-descentes $[K_I^*]\{v_{IK}\} = \{G_{Ik}^*\}$
Nouvelle BOUCLE SUR LES SOUS-STRUCTURES n=1,N
Restitution des nœuds internes $[v_{\Omega k}^n]$
Construction de la solution globale $[v_k]$
Fin de la BOUCLE SUR LES SOUS-STRUCTURES n=1,N
Fin de la BOUCLE SUR LES ORDRES k=1,p

Figure 3. *Algorithme proposé*

$NNPE$ *étant le nombre de nœuds par élément*

$NDPN$ *étant le nombre de degrés de liberté par nœud,*

soit :

$$3(N+1) * N_\Omega * \beta * N_I^2.$$

Le nombre total d'opérations pour la construction de $[K_I^*]$ par notre algorithme est de :

$$(N+1) * N_I * N_\Omega(4N_{dim} + 3\beta * N_I) + N * N_\Omega * N_{dim}^2.$$

Pour illustrer cette dernière formule, considérons d'abord le cas d'un raffinement de maillage dans la largeur. Le nombre $N.N_\Omega$ est à peu près le nombre de degrés de liberté total. Ce nombre $N.N_\Omega$, ainsi que le nombre de degrés de liberté d'interface N_I et que N_{dim}, augmente proportionnellement au nombre de degrés de liberté N_{DDL}.

Le temps de calcul de l'interface augmente proportionnellement au cube de N_{DDL} comme pour la triangulation d'une méthode directe. Donc dans ce cas de raffinement en largeur, la sous-structuration en longueur ne semble pas apporter d'amélioration en temps de calcul. Au contraire, pour un raffinement selon la longueur, seul le produit $N.N_\Omega$ augmente, N_I et N_{dim} restant fixés. Dans ce cas, le temps de calcul de la matrice d'interface augmente proportionnellement à N_{DDL}.

7. Un test numérique et discussion

Nous allons appliquer notre algorithme de sous-structuration au problème de la compression d'une plaque élastique, sous forme d'une bande. On se propose de calculer le pré et post-flambage de cette plaque en description lagrangienne totale. Le module d'Young du matériau est $E = 2.10^5 Mpa$, son coefficient de Poisson est $\nu = 0.3$, sa longueur $133\ mm$ et l'on a un rapport d'aspect égal à 33.25, ce qui correspond à une plaque longue. Cette plaque est encastrée d'un côté et soumise à une force de compression uniforme de l'autre côté, (voir figure 4).

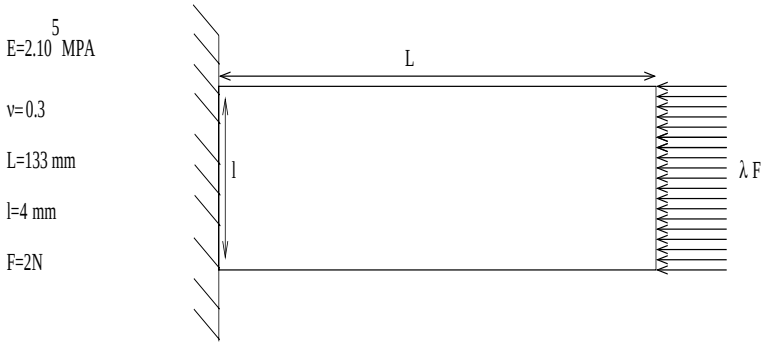

Figure 4. *Plaque encastrée sur un côté et soumise à une compression axiale*

La plaque est discrétisée en éléments triangulaires DKT18 [BAT 90]. Nous présentons une étude détaillée des temps de calculs obtenus avec la MAN pour un problème résolu avec ou sans sous-structuration, notons que l'on a choisi un ordre de troncature $p = 20$ et une précision de $\epsilon = 10^{-6}$.

Dans un premier temps, nous avons choisi d'utiliser cinq maillages de dimensions différentes (3990 d.d.l, 12000 d.d.l, 24000 d.d.l, 50000 d.d.l, 100000 d.d.l), tout en gardant le même nombre de nœuds, égal à 5, suivant la largeur de la plaque.

Pour évaluer l'algorithme proposé, nous avons également fait varier le nombre de sous-structures (3, 6, 12, 18, 22, 33). Les temps CPU relatifs aux opérations de construction de $[K_I^*]$, de sa décomposition ainsi que de la construction des seconds membres $\{G_{Ik}^*\}$ sont donnés dans les tableaux (1), (2) et (3). Dans tous les cas, le temps de triangulation de la matrice d'interface $[K_I^*]$ est négligeable, car elle est souvent de petite taille et avec une faible largeur de bande, due à la forme de la structure.

Donc le temps de calcul se partage entre le calcul des vecteurs seconds membres $\{G_I^*\}$ et le calcul de la matrice d'interface. On remarque que le coût de calcul des seconds membres dépend peu de la sous-structuration et qu'il est proportionnel au nombre de degrés de liberté. Cela veut dire que la sous-structuration n'alourdit pas le calcul des seconds membres qui est essentiel dans l'application de la MAN. On constate ensuite que la condensation des nœuds internes demande beaucoup de temps de calcul, ce qui est bien connu, mais ici le temps reste inférieur au calcul des 20 seconds membres. Plus précisément, le rapport est presque égal à un, pour de gros sous-domaines, alors qu'il peut être très inférieur à $1/2$ pour de petites sous-structures. En comparant les lignes successives des tableaux 1 à 3, on voit que ce temps de calcul de la matrice d'interface évolue comme $(N_{DDL})^{\alpha}$ avec un exposant α voisin de 1, conformément à l'analyse présentée précédemment.

	3				6			
	calcul $[K_I^*]$	décomp $[K_I^*]$	Calcul $\{G_I^*\}$	Total T	calcul $[K_I^*]$	décomp $[K_I^*]$	Calcul $\{G_I^*\}$	Total T
3990 ddl	4.4	.02	16.3	21	4.5	.01	16.8	20
12000 ddl	22.8	.01	53.2	74	20.4	.01	51.2	71
24000 ddl	41.3	.01	106.6	147	59.3	.01	103.5	162
50000 ddl	120.3	.03	214.3	334	134.6	.01	205	339
100000 ddl	351	.01	426.1	777	320	0.04	411	731

Tableau 1. *Détails des temps CPU de la MAN avec sous-structuration. Raffinement de maillage dans la longueur*

	12				18			
	calcul $[K_I^*]$	décomp $[K_I^*]$	Calcul $\{G_I^*\}$	Total T	calcul $[K_I^*]$	décomp $[K_I^*]$	Calcul $\{G_I^*\}$	Total T
3990 ddl	5.7	.01	16.3	21	6.2	.01	17.8	23
12000 ddl	28.5	.001	49.8	7	15.3	.03	53.82	687
24000 ddl	61.1	.01	102.1	164	45.8	.02	108.9	154
50000 ddl	136.1	.01	208.8	343	99.2	.04	219.3	318
100000 ddl	230	.05	433	663	160	.07	456.31	616

Tableau 2. *Détails des temps CPU de la MAN avec sous-structuration. Raffinement de maillage dans la longueur*

Sur la figure (5), nous avons presenté l'evolution du temps CPU total en fonction du nombre de sous-structures pour l'exemple de 100 000 degrés de liberté. Pour cet exemple, l'optimum est obtenu avec un assez grand nombre de sous-domaines, entre 15 et 25, mais le temps CPU dépend assez peu de la sous-structuration, à cause de l'influence prépondérante des seconds membres.

Les temps nécessaires aux mêmes calculs mais cette fois-ci sans sous-structuration sont donnés dans le tableau 4. Sur cet exemple, la décomposition ne coûte presque

	22				33			
	calcul $[K_I^*]$	décomp $[K_I^*]$	Calcul $\{G_I^*\}$	Total T	calcul $[K_I^*]$	décomp $[K_I^*]$	Calcul $\{G_I^*\}$	Total T
3990 ddl	5.8	.05	22.6	27	6	.06	24.1	30
12000 ddl	17.3	.06	55.2	71	14.2	.07	58.7	72
24000 ddl	34.3	.04	112.1	146	62.3	.06	120.8	182
50000 dd	87.9	.05	227.2	314	l74.44	.07	241	415
100000 ddl	140	.08	480	620	212	.08	514.3	726

Tableau 3. *Détails des temps CPU de la MAN avec sous-structuration. Raffinement de maillage dans la longueur*

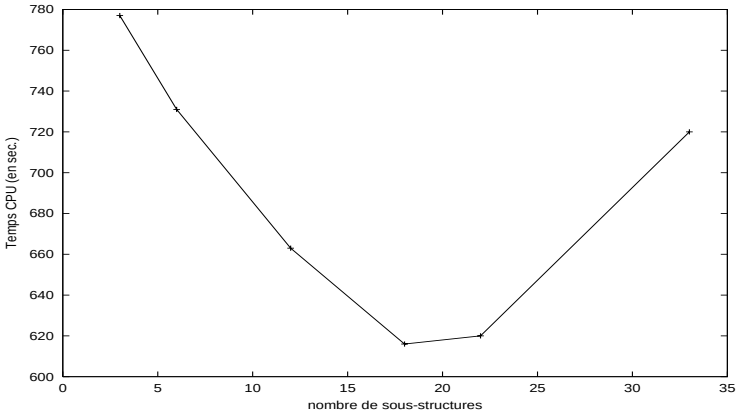

Figure 5. *Evolution du temps CPU total en fonction du nombre de sous-structures pour l'exemple de 100 000 d.d.l*

rien, à cause de la forme de la structure, qui induit une faible largeur de bande. Rappellons que dans la plupart des analyses faites sur la MAN, le coût de la décomposition est au moins égal au calcul de 15 ou 20 seconds membres. A grand nombre d'inconnues, les coûts de calcul obtenus ne sont que légèrement inférieurs à ceux de la MAN avec sous-structuration.

	Calcul $[K]$	Decomposition $[K]$	Calcul $\{SM\}$	CPU Total
3990 ddl	2.6	0.2	18.5	23.9
12000 ddl	7.82	0.6	56.1	73.2
24000 ddl	15.6	1.4	112.5	134.4
50000 ddl	31.4	2.7	226.5	265
100000 ddl	62.5	5.5	453.4	520

Tableau 4. *Détails des temps CPU de la MAN sans sous-structuration*

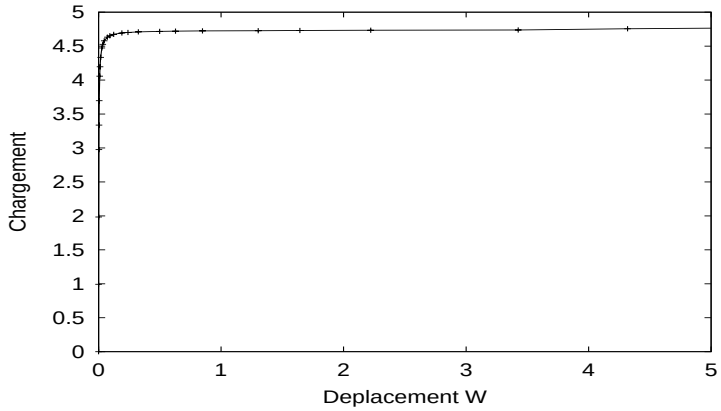

Figure 6. *Courbe charge déplacement, MAN avec et sans sous-structuration, 10 pas, ordre 20*

Pour évaluer l'influence de la largeur de bande sur le temps d'exécution des différentes opérations, nous avons conservé le même exemple et nous avons choisi cette fois-ci cinq maillages de dimensions différentes (3 990, 7 990, 11 970, 15 990, 23 940) en ne changeant que le nombre de nœuds suivant la largeur. Nous avons retenu un découpage en 3 sous-structures pour les cinq maillages. Nous présentons les temps CPU detaillés relatifs aux opérations de construction de $[K_I^*]$, de sa décomposition ainsi que de la construction des seconds membres $\{G_{Ik}^*\}$ dans le tableau 5. Comme cela avait été prévu, c'est le calcul des matrices $[P_1]$ et $[P_2]$ qui est le plus coûteux avec un assez grand nombre de degrés de liberté et le temps CPU augmente presque comme le cube du nombre d'inconnues. Cela induit des coûts de calcul assez importants pour les plus grands cas testés (16 000, 24 000 ddl), mais la différence reste acceptable par rapport à la méthode sans sous-structuration. Notons que la croissance modérée du temps observée au tableau 6 est due au faible coût de la triangulation. Toutefois, quelle que soit la forme de la structure étudiée, il semble déconseillé de calculer explicitement la matrice d'interface lorsque les sous-domaines sont trop importants.

Egalement, les temps CPU nécessaires aux mêmes opérations, mais cette fois-ci sans sous-structuration, sont donnés dans le tableau 6.

Les résultats obtenus par la MAN avec ou sans sous-structuration sont identiques (voir figure 6). La qualité de la solution est la même pour les deux types de résolution, les valeurs des domaines de validité sont exactement les mêmes, quelle que soit la méthode utilisée (avec ou sans sous-structuration). Ces résultats identiques obtenus avec les deux méthodes s'expliquent par le choix du solveur utilisé pour résoudre les problèmes [12]. En effet nous utilisons un solveur direct ce qui conduit à une résolution exacte des problèmes linéaires successifs. Une résolution itérative de ces systèmes demanderait de faire un choix sur la précision du solveur itératif, ce qui conduirait alors à une différence dans les résultats obtenus avec la MAN sans sous-structuration.

	Calcul $[K*]$			Decomposition $[K*]$	Calcul $\{SM\}$	CPU Total
	kij	P1 et P2	autres produits			
3990 ddl	2.4	1.92	.05	.02	16.39	21
7980 ddl	4	14.8	0.2	.09	40.2	59
11970 ddl	9.8	55.7	0.53	0.15	68	133.5
15960 ddl	18.15	132.51	0.98	0.35	99.45	250.6
23940 ddl	29.61	256.92	1.52	0.68	133.6	421.6

Tableau 5. *Détails des temps CPU de la MAN sans sous-structuration. Raffinement suivant la largeur*

	Calcul $[K]$	Decomposition $[K]$	Calcul $\{SM\}$	CPU Total
3990 ddl	2.6	0.2	18.5	23.9
7980 ddl	6.22	1.32	42.6	50.14
11970 ddl	9.9	3.89	68.80	81.6
15960 ddl	13.3	8.2	96.9	118
23940 ddl	16.6	15.4	128.6	160.6

Tableau 6. *Détails des temps CPU de la MAN sans sous-structuration. Raffinement suivant la largeur*

Galliet [GAL 00] a montré, dans sa thèse, que cette résolution itérative devait se faire avec une précision égale ou inférieure à la précision souhaitée sur la qualité de la solution. Contrairement aux applications précédentes de la MAN, le temps prédominant est celui du calcul des seconds membres quelle que soit la méthode utilisée (avec ou sans sous-structuration).

En résumé, on a montré que le calcul explicite d'une matrice d'interface restait du même ordre que le calcul des seconds membres de la Méthode Asymptotique Numérique, à condition que les matrices de sous-domaines $[K^n_\Omega]$ ne soient pas trop grosses et que la largeur de bande reste raisonnable. Les calculs des seconds membres sont en fait des calculs d'intégrales, analogues à des calculs de résidus, et donc naturellement parallélisables. On sait qu'il n'en est pas de même pour l'étape de triangulation d'une matrice par une méthode directe classique (Gauss, Choleski, Crout), qui est nécessairement séquentielle. La décomposition en sous-domaine permet de remplacer la triangulation globale par les calculs de condensation de la formule [11], qui peuvent être répartis par sous-domaine. L'introduction de la sous-structuration permet donc de restreindre fortement la part des calculs non parallélisables. Par ailleurs, cela permet de réduire le nombre de réels à stocker.

8. Conclusion et perspectives

Nous avons mis au point un nouvel algorithme basé sur l'association de la méthode de sous-structuration et de la MAN pour le calcul non linéaire des plaques sous forme

de bande. Les résultats obtenus sont identiques à ceux obtenus avec une MAN sans sous-structuration. Ces résultats montrent également que la programmation séquentielle de l'algorithme proposé est relativement proche, en temps de calcul, à la version MAN sans sous-structuration. Les opérations les plus coûteuses en temps de calcul se révèlent être les opérations de calcul de l'opérateur condensé $[K_I^*]$ et des seconds membres $\{G_I^*\}$. Ces opérations sont facilement parallélisables et doivent conduire à des diminutions conséquentes des temps de calculs, avec un ordinateur à architecture parallèle.

Naturellement, l'implémentation de cet algorithme sur un tel ordinateur est l'une des perspectives de ce travail, ainsi qu'une comparaison avec d'autres méthodes de couplage de la MAN avec la sous-structuration. Puisque le calcul de la matrice d'interface peut être coûteux à grand nombre d'inconnues, une autre possibilité est de faire une condensation une ou deux fois seulement et de se servir de cette matrice comme préconditionneur pour des résolutions ultérieures.

Signalons enfin que cette méthode reste valable pour des sous-domaines touchant à plus de deux autres sous-domaines et ne devrait affecter que la topologie de la matrice d'interface.

Remerciements

Ce travail a été financé par le Centre National Marocain de la Recherche Scientifique et Technique (PARS, SPI :016).

9. Bibliographie

[BAT 90] BATOZ J. L., DHATT G., *Modélisation des structures par Éléments Finis*, Hermes, VOL. 3, 1990.

[BAR 99] BARBOTEU M., Contact, Frottement et Techniques de Calcul Parallèle , Thèse, Université Montpellier II, 1999.

[COC 941] COCHELIN B., "A path-following technique via an asymptotic-numerical method", *Computers and structures*, VOL. 53, P. 1181–1192, 1994.

[COC 942] COCHELIN B., DAMIL N., POTIER-FERRY M., "The asymptotic-numerical method : an efficient perturbation technique for nonlinear structural mechanics", *Revue Européenne des Eléments Finis*, VOL. 3, P. 281–297, 1994.

[COC 943] COCHELIN B., DAMIL N., POTIER-FERRY M., Asymptotic-Numerical Methods and Padé approximants for non-linear elastic structures, *International Journal of Numerical Methods in Engineering*, VOL. 37, P. 1187-1237, 1994.

[DAD 80] DODDS R.H.JR., LOPEZ L.A., Substructuring in linear and non linear analysis, *International Journal of Numerical Methods in Engineering*, VOL. 15, P. 583-597, 1980.

[DES 90] DESTUYNDER P., OUSSET Y., Quelques méthodes de sous-structuration dynamique, *Calcul des structures et intelligence artificielle*, VOL. 3, P. 377-402, 1990.

[ERH 90] ERHEL J., VIDRASCU M., Adaptation aux super-ordinateurs des algorithmes et structures de données spécifiques des éléments finis, *Calcul des structures et intelligence artificielle*, VOL. 3, P. 403-417, 1990.

[ESC 92] ESCAIG Y., Décomposition de domaine multiniveaux et traitements distribués pour la résolution de problèmes de grande taille, Thèse de doctorat, Université de Technologie de Compiègne, 1992.

[ESC 99] ESCAIG Y., MARIN P., Exemples of domain decomposition methods to solve non-linear problems sequentially, *Advances in Engineering Software*, VOL. 30, P. 847-855, 1999.

[FAR 941] FARHAT C., GERADIN M., On a component mode synthesis method and its application to incompatible substructure, *Computer and Structures*, VOL. 51, P. 459-473, 1994.

[FAR 942] FARHAT C., ROUX F.X., Implicit parallel processing in structural Mechanics, *Computer Mechanics Advances*, VOL. 2, P. 47-72, 1994.

[GAL 00] GALLIET I., Une version parallèle des méthodes asymptotiques numériques. Applications à des structures complexes à base d'élastomères., Thèse de doctorat, Université Marseille II, 2000.

[MEO 02] MEO S., DÉBORDES O., BOUKAMEL A., Assemblage de structures une à une invariantes dans une direction, *Mécanique et Industries*, VOL. 3, P. 211–225, 2002.

[PRE 63] PREZEMIENIESKI J. S., Matrix structural anlysis of substructures, *Am. Inst.Aero. Astro. J.* , VOL. 1, P. 138–147, 1963.

[REY 96] REY C., Une technique d'accélération de la résolution de problèmes d'élasticité non linéaire par décomposition de domaines, *Compte-Rendus de l'Académie des Sciences. Paris.*, VOL. T.322, SÉRIE IIB, P. 601–606, 1996.

[ROU 90] ROUX F. X., Méthodes de résolution par sous-domaines en statique, *Calcul des structures et intelligence artificielle* , VOL. 3, P. 353–376, 1990.

[SCH 69] SCHWARTZ H.A., Uber einige Abbildungsaufgaben, *Gesammelte Mathematische Abhandlungen* , VOL. 11, P. 65–83, 1869.

[TAL 93] LE TALLEC P., VIDRASCU M., Méthodes de décomposition de domaines en calcul des structures , *Premier Colloque National en Calcul des Structures. Giens.* 1993.

[WES 92] WESSELING P., *An introduction to multigrid methods*, John Wiley and Sons, 1992.

Annexe

La deuxième équation du problème [8] nous permet d'éliminer les inconnues intérieures à la sous-structure $\Omega(n)$ $\{v^n_{\Omega k}\}$, elle est sous la forme :

$$\begin{aligned}\{v^n_{\Omega\ k}\} \quad &= \lambda_k [\ K^n_\Omega]^{-1}\{F^n_\Omega\} - [\ K^n_\Omega]^{-1}[\ K^{n,n}_c]\{v^n_{Ik}\} \\ &-[\ K^n_\Omega]^{-1}[\ K^{n-1,n}_c]^t\{v^{n-1}_{Ik}\} + [\ K^n_\Omega]^{-1}\{FQ^n_{\Omega k}\}.\end{aligned} \qquad [17]$$

En reportant [17] dans la troisième équation de [8], on trouve :

$$
\begin{aligned}
([K_I^{n,n}] - [K_c^{n,n}]^t[K_\Omega^n]^{-1}[K_c^{n,n}])\{v_{Ik}^n\} - [K_c^{n,n}]^t[K_\Omega^n]^{-1}[K_c^{n-1,n}]^t\{v_{Ik}^{n-1}\} \\
= -\lambda_k[K_c^{n,n}]^t[K_\Omega^n]^{-1}\{F_\Omega^n\} + \lambda_k\{F_I^{n,n}\} + \{f_k^{n+1,n}\} \\
-[K_c^{n,n}]^t[K_\Omega^n]^{-1}\{FQ_{\Omega k}^n\} + \{FQ_{Ik}^{n,n}\}.
\end{aligned}
\quad [18]
$$

D'une part, l'équilibre de la sous structure $\Omega(n+1)$ donne une équation analogue à l'équation [18] couplant les vecteurs $\{f_k^{n,n+1}\}$,$\{v_{Ik}^n\}$,$\{v_{Ik}^{n+1}\}$, $\{F_\Omega^{n+1}\}$, $\{F_I^{n+1,n}\}$, $\{FQ_{\Omega k}^{n+1}\}$,$\{FQ_{Ik}^{n+1,n}\}$.

D'autre part, le principe de l'action et de la réaction, appliqué à chaque ordre k, au niveau de chaque interface s'écrit :

$$\{f_k^{n+1,n}\} + \{f_k^{n,n+1}\} = 0$$

ce qui nous permet d'obtenir le problème d'interfaces suivant :

$$
\begin{aligned}
&\big([K_I^{n,n}] - [K_c^{n,n}]^t[K_\Omega^n]^{-1}[K_c^{n,n}] + [K_I^{n,n+1}] \\
&-[K_c^{n,n+1}][K_\Omega^{n+1}]^{-1}[K_c^{n,n+1}]^t\big)\{v_{Ik}^n\} - [K_c^{n,n+1}][K_\Omega^{n+1}]^{-1}[K_c^{n+1,n+1}]\{v_{Ik}^{n+1}\} \\
&-[K_c^{n,n}]^t[K_\Omega^n]^{-1}[K_c^{n-1,n}]^t\{v_{Ik}^{n-1}\} \\
&= [K_c^{n,n}]^t[K_\Omega^n]^{-1}(\lambda_k F_\Omega^n + FQ_{\Omega\ k}^n) - [K_c^{n,n+1}][K_\Omega^{n+1}]^{-1}(\lambda_k F_\Omega^{n+1} + FQ_{\Omega\ k}^{n+1}) \\
&+\lambda_k(F_I^{n,n} + F_I^{n+1,n}) + FQ_{Ik}^{n+1,n} + FQ_{Ik}^{n,n}.
\end{aligned}
\quad [19]
$$

Après assemblage, à chaque ordre k $(k = 1, p)$, de tous les problèmes [19] en faisant varier n de 1 à N, on obtient le problème condensé à l'ordre k suivant :

$$
\begin{bmatrix}
[K_I^{*(0)}] & [K_c^{*(0,1)}] & [0] & . & . & . & [0] \\
[K_c^{*(0,1)}]^t & [K_I^{*(1)}] & [K_c^{*(1,2)}] & [0] & . & . & [0] \\
[0] & [K_c^{*(1,2)}]^t & [K_I^{*(2)}] & [K_c^{*(2,3)}] & . & . & [0] \\
. & . & . & . & . & . & . \\
. & . & . & . & . & . & . \\
. & & . & . & . & . & . \\
. & & . & . & . & . & . \\
. & & & . & . & . & . \\
[0] & . & . & [0] & . & [K_c^{*(N-1,N)}]^t & [K_I^{*(N)}]
\end{bmatrix}
\begin{Bmatrix}
u_{Ik}^0 \\ u_{Ik}^1 \\ . \\ . \\ . \\ . \\ . \\ . \\ u_{Ik}^N
\end{Bmatrix}
=
\begin{Bmatrix}
g_{Ik}^{*(0)} \\ g_{Ik}^{*(1)} \\ . \\ . \\ . \\ . \\ . \\ . \\ g_{Ik}^{*(N)}
\end{Bmatrix}.
\quad [20]
$$

Nous présentons, sur la figure suivante, un résumé des étapes proposées dans notre travail.

étape 1 : décomposition de la structure Ω

$$\Omega = \cup_{i=1}^{N} \Omega(n)$$

$$\Downarrow$$

étape 2 : Application de la MAN pour chaque $\Omega(n)$

$$[K_t^n]\{v_k^n\} = \lambda_k\{F^n\} + \{f_k^{n-1,n}\} + \{f_k^{n+1,n}\} + \{FQ_k^n\} \quad n = 1, ..., N$$

pour n=1,N et k=1,p

$$\Downarrow$$

étape 3 : condensation sur I(n)

pour n=1,N et k=1,p

$$[K_I^n]\{v_{Ik}^{n-1,n,n+1}\} = \big([K_I^{n,n}] - [K_c^{n,n}]^t[P_1] + [K_I^{n,n+1}] - [K_c^{n,n+1}][P_2]\big)\{v_{Ik}^n\} - [P_2]^t[K_c^{n+1,n+1}]\{v_{Ik}^{n+1}\} - [P_1]^t[K_c^{n-1,n}]^t\{v_{Ik}^{n-1}\}$$

$$\{g_{Ik}^n\} = [P_1]^t(\lambda_k\{F_\Omega^n\} + \{FQ_{\Omega\,k}^n\}) - [P_2]^t(\lambda_k\{F_{\Omega\,k}^{n+1}\} + \{FQ_{\Omega\,k}^{n+1}\}) + \lambda_k(\{F_I^{n,n}\} + \{F_I^{n+1,n}\}) + \{FQ_{Ik}^{n+1,n}\} + \{FQ_{Ik}^{n,n}\}$$

$$\Downarrow$$

étape 4 : assemblage

$$[K_I^n] \rightarrow [K_I^*]$$
$$\{g_{Ik}^n\} \rightarrow \{G_{Ik}^*\}$$

$$\Downarrow$$

étape 5 : résolution du problème d'interface

$$\begin{cases} \textit{Trouver} \ \{u_{Ik}\} \ \textit{tel que} \\ [K_I^*]\{u_{Ik}\} = \{G_{Ik}^*\} \end{cases}$$

$$\Downarrow$$

étape 6 : restitution des solutions sur $\Omega(n)$

Pour n=1,N et k=1,p

$$\{v_{\Omega\,k}^n\} = \lambda_k[K_\Omega^n]^{-1}\{F_\Omega^n\} + [K_\Omega^n]^{-1}\{FQ_\Omega^n\} \ \ pour \ \ n = 1, N$$
$$-[K_\Omega^n]^{-1}[K_c^{n-1,n}]^t\{v_{Ik}^{n-1}\} - [K_\Omega^n]^{-1}[K_c^{n,n}]^t\{v_{Ik}^n\}$$

Figure 7. *Détail des différentes étapes du couplage de la sous-structuration et la MAN*